



COMPTE RENDU

**Détection de langue de rédaction d'un texte en  
utilisant le concept de traitement naturel du langage**

Souheib Ben Mabrouk

Encadré par : Mme. Sofia Ben Jebara

INDP2E

1<sup>er</sup> février 2022

# Table des figures

1	Jeu de données : Language_detection . . . . .	2
2	Les 10 premières lignes des données . . . . .	3
3	Pourcentage de chaque langue en jeux de données . . . . .	3
4	Les valeurs prises par <b>y</b> après encodage de la variable ‘Langue’ . . . . .	4
5	Exemples de 10 textes de jeu de données avant et après traitement . . . . .	6
6	Exemple illustartif des Sac de mots . . . . .	7
7	Matrice confusion du modèle étudié . . . . .	9
8	Résultat de prédiction des langues des échantillons données . . . . .	10

## I. Objectif

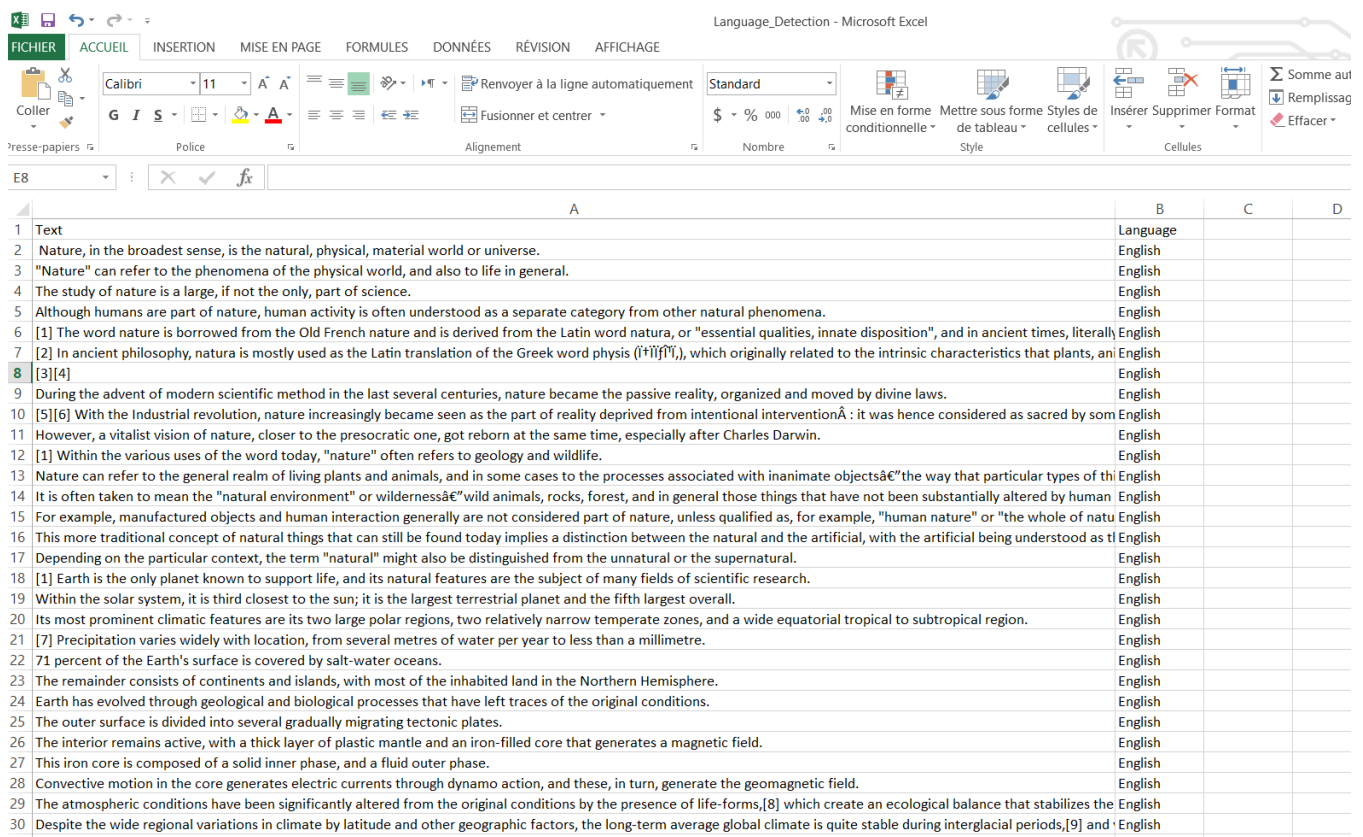
Dans ce TD machine, on se propose de développer une solution d'apprentissage machine (ML) permettant la détection de la langue d'un texte écrit en se basant sur les concepts du traitement du langage naturel (natural language processing (NLP)).

Pour ce faire, j'ai choisi de travailler avec **Python 3.9** par le recours à **Jupyter Notebook** qui me facilite le travail grâce à sa simplicité et son efficacité en termes de partage du code en un ensemble de blocs. Il offre un environnement interactif et facile à utiliser, qui ne fonctionne pas seulement comme un environnement de développement intégré (IDE), mais aussi comme un outil de présentation ou d'enseignement. [1]

## II. Dataset :

On utilisera un ensemble de données de texte écrits en 17 langues différentes, c'est-à-dire qu'on créera un modèle NLP pour prédire **17 langues différentes**.

Il s'agit des langues suivantes : Allemand, Anglais, Arabe, Danois, Espagnol, Français, Grec, Hindi , Italien, Kannada , Malayalam , Néerlandais, Portugais, Russe, Suédois, Tamoul et Turc. Le fichier 'language\_detection.csv' contient le même jeu de données composé de 10 267 données organisées.



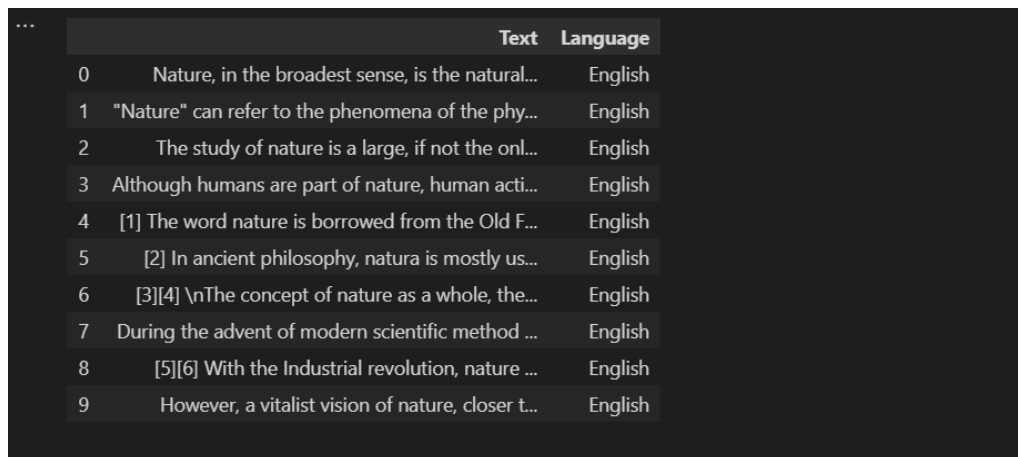
	A	B	C	D
1	Text	Language		
2	Nature, in the broadest sense, is the natural, physical, material world or universe.	English		
3	"Nature" can refer to the phenomena of the physical world, and also to life in general.	English		
4	The study of nature is a large, if not the only, part of science.	English		
5	Although humans are part of nature, human activity is often understood as a separate category from other natural phenomena.	English		
6	[1] The word nature is borrowed from the Old French nature and is derived from the Latin word natura, or "essential qualities, innate disposition", and in ancient times, literally	English		
7	[2] In ancient philosophy, natura is mostly used as the Latin translation of the Greek word physis (φύσις), which originally related to the intrinsic characteristics that plants, ani	English		
8	[3][4]	English		
9	During the advent of modern scientific method in the last several centuries, nature became the passive reality, organized and moved by divine laws.	English		
10	[5][6] With the Industrial revolution, nature increasingly became seen as the part of reality deprived of intentional intervention: it was hence considered as sacred by som	English		
11	However, a vitalist vision of nature, closer to the presocratic one, got reborn at the same time, especially after Charles Darwin.	English		
12	[1] Within the various uses of the word today, "nature" often refers to geology and wildlife.	English		
13	Nature can refer to the general realm of living plants and animals, and in some cases to the processes associated with inanimate objects: "the way that particular types of thi	English		
14	It is often taken to mean the "natural environment" or wilderness: "wild animals, rocks, forest, and in general those things that have not been substantially altered by human	English		
15	For example, manufactured objects and human interaction generally are not considered part of nature, unless qualified as, for example, "human nature" or "the whole of natu	English		
16	This more traditional concept of natural things that can still be found today implies a distinction between the natural and the artificial, with the artificial being understood as t	English		
17	Depending on the particular context, the term "natural" might also be distinguished from the unnatural or the supernatural.	English		
18	[1] Earth is the only planet known to support life, and its natural features are the subject of many fields of scientific research.	English		
19	Within the solar system, it is third closest to the sun; it is the largest terrestrial planet and the fifth largest overall.	English		
20	Its most prominent climatic features are its two large polar regions, two relatively narrow temperate zones, and a wide equatorial tropical to subtropical region.	English		
21	[7] Precipitation varies widely with location, from several metres of water per year to less than a millimetre.	English		
22	71 percent of the Earth's surface is covered by salt-water oceans.	English		
23	The remainder consists of continents and islands, with most of the inhabited land in the Northern Hemisphere.	English		
24	Earth has evolved through geological and biological processes that have left traces of the original conditions.	English		
25	The outer surface is divided into several gradually migrating tectonic plates.	English		
26	The interior remains active, with a thick layer of plastic mantle and an iron-filled core that generates a magnetic field.	English		
27	This iron core is composed of a solid inner phase, and a fluid outer phase.	English		
28	Convective motion in the core generates electric currents through dynamo action, and these, in turn, generate the geomagnetic field.	English		
29	The atmospheric conditions have been significantly altered from the original conditions by the presence of life-forms,[8] which create an ecological balance that stabilizes the	English		
30	Despite the wide regional variations in climate by latitude and other geographic factors, the long-term average global climate is quite stable during interglacial periods,[9] and	English		

Fig. 1 – Jeu de données : Language\_detection

On commence par importer la bibliothèque **pandas** de gestion des dataframes. Puis, on lit le jeu de données et on observe le contenu des 10 premières lignes :

```
1 import pandas as pd
2 data = pd.read_csv("files/Language_Detection.csv")
3 data.head(10)
```

On obtient le résultat suivant :



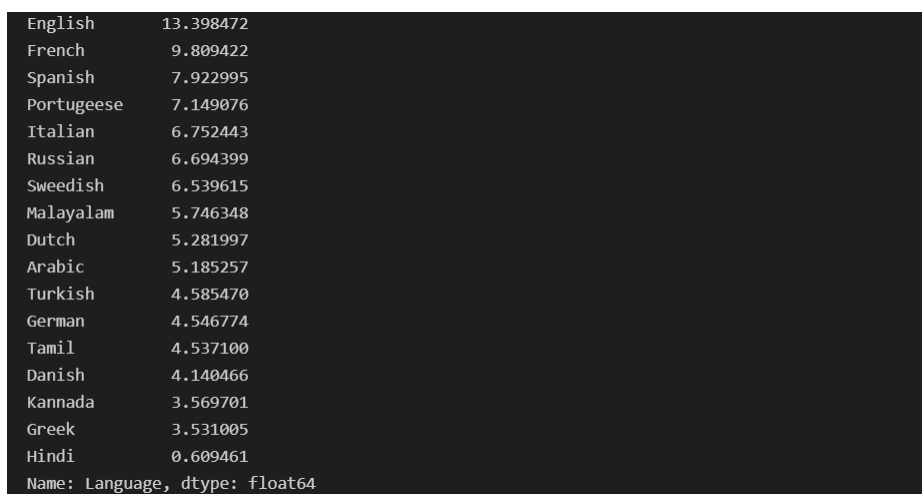
	Text	Language
0	Nature, in the broadest sense, is the natural...	English
1	"Nature" can refer to the phenomena of the phy...	English
2	The study of nature is a large, if not the onl...	English
3	Although humans are part of nature, human acti...	English
4	[1] The word nature is borrowed from the Old F...	English
5	[2] In ancient philosophy, natura is mostly us...	English
6	[3][4] \nThe concept of nature as a whole, the...	English
7	During the advent of modern scientific method ...	English
8	[5][6] With the Industrial revolution, nature ...	English
9	However, a vitalist vision of nature, closer t...	English

**Fig. 2** – Les 10 premières lignes des données

**On remarque qu'il s'agit d'un fichier structuré de données formé de plusieurs échantillons (lignes) et de 2 attributs par échantillon (colonnes).**

On peut compter le nombre de données de chaque langue et son pourcentage comme suit :

```
1 p=data["Language"].value_counts()
2 p=p/len(data)*100
3 p
```



English	13.398472
French	9.809422
Spanish	7.922995
Portugeese	7.149076
Italian	6.752443
Russian	6.694399
Sweedish	6.539615
Malayalam	5.746348
Dutch	5.281997
Arabic	5.185257
Turkish	4.585470
German	4.546774
Tamil	4.537100
Danish	4.140466
Kannada	3.569701
Greek	3.531005
Hindi	0.609461

Name: Language, dtype: float64

**Fig. 3** – Pourcentage de chaque langue en jeux de données

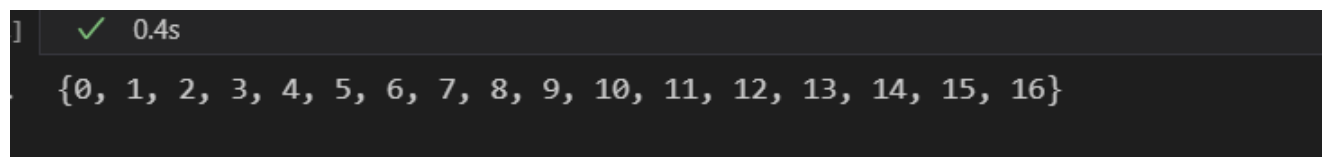
### III. Séparation des variables et encodage de "Langue"

Maintenant, nous pouvons séparer les variables dépendantes et indépendantes. Ici, les données de texte constituent la variable indépendante tandis que le nom de la langue est la variable dépendante et **catégorielle**. Pour entraîner le modèle de détection de la langue, elle doit être convertie sous une forme **numérique**.

On procède le démarche suivant :

```
1 X = data["Text"]
2 y = data["Language"]
3 from sklearn.preprocessing import LabelEncoder
4 le = LabelEncoder()
5 y = le.fit_transform(y)
6 set(y) #permet d'afficher les valeurs possibles de y sans redondance
```

On obtient le résultat suivant :



```
] ✓ 0.4s
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
```

**Fig. 4** — Les valeurs prises par **y** après encodage de la variable 'Langue'

On remarque que les valeurs catégorielles sont transformées en des valeurs numériques allant de 0 à 16 pour désigner les 17 différentes langues dans l'ordre alphabétique :

Valeur numérique associée	Langue
0	Arabic
1	Danish
2	Dutch
3	English
4	French
5	German
6	Greek
7	Hindi
8	Italian
9	Kannada
10	Malayalam
11	Portugeese
12	Russian
13	Spanish
14	Sweedish
15	Tamil
16	Turkish

## IV. Prétraitement du texte

L'ensemble de données utilisé a été obtenu par **Web scrapping**<sup>1</sup>. Cependant, ces textes contiennent de nombreux symboles indésirables et des nombres qui peuvent affecter la qualité du modèle. Pour cela, nous remplacerons tous les symboles indisérables par des espaces pour tous les textes du jeu de données. On procède par le démarche suivant :

```

1 import re
2 data_list = []
3 for text in X:
4     text = re.sub(r'[@#$(,n"%^*?:;~`0-9]', ' ', text)
5     text = re.sub(r'[][]', ' ', text)
6     text = text.lower()
7     data_list.append(text)

```

---

1. L'art d'extraire des données depuis un site web a un nom : c'est le web scraping, aussi appelé harvesting. Cette technique permet de récupérer des informations d'un site, grâce à un programme ou un logiciel et de les réutiliser ensuite. En automatisant ce process, on évite ainsi de devoir récolter les données manuellement, on gagne du temps et on accède à un fichier unique et structuré.[2]

```

8 for i in range(10):
9     print(X[1000*i])
10    print(data_list[1000*i])
11    print("-----")

```

**Le pré-traitement réalisé assure la transformation des caractères spéciaux indisérables par des espaces, la transformation de la lettre "n" par une espace aussi (car elle figure dans la liste des caractères à remplacer). De plus, toutes les lettres sont mises en miniscule et les caractères [ ] sont conservés ! De plus, On garde les "."**

**On visualise ainsi l'effet du prétraitement sur les lignes {1,1001,2001,...,9001} (afin de voir l'effet pour différentes langues) tout en ajoutant des lignes de séparation pour mieux visualiser le résultat :**

```

Nature, in the broadest sense, is the natural, physical, material world or universe.
nature i the broadest se se is the atural physical material world or u iverse.
-----
When trained on man-made data, machine learning is likely to pick up the same constitutional and unconscious biases already present in so
whe trai ed o ma -made data machi e lear i g is likely to pick up the same co stitutio al a d u co scious biases already prese t i so
-----
वाक्यांश संख्या दो, जब आपने किसी को कुछ समय के लिए नहीं देखा है, तो आप उसे बताएंगे कि कोई समय नहीं है, तो इसका मतलब है कि आपने इस व्यक्ति को शायद हफ्ते
वाक्यांश संख्या दो जब आपने किसी को कुछ समय के लिए नहीं देखा है तो आप उसे बताएंगे कि कोई समय नहीं है तो इसका मतलब है कि आपने इस व्यक्ति को शायद हफ्ते
-----
Meu Deus.
meu deus.
-----
merci d'avance si quelqu'un a fait une erreur, et il en est vraiment énervé.
merci d'ava ce si quelqu'u a fait u e erreur et il e est vraïme t é ervé.
-----
Con excepció de ciertas personas remuneradas por la Fundación Wikimedia,[131] el resto, conocidos en la jerga de Wikipedia como wikipedi
co excepció de ciertas perso as remu eradas por la fu dació wikimedia [ ] el resto co ocidos e la jerga de wikipedia como wikipedi
-----
Журнал ориентирован на научных работников, однако в начале каждого издания публикуется краткое популярное изложение важнейших публикаций.
журнал ориентирован на научных работников однако в начале каждого издания публикуется краткое популярное изложение важнейших публикаций.
-----
lad mig afslutte.
lad mig afslutte.
-----
yani biriyle tanışın ve zamanın nasıl gidiyor?
...
ya i biriyle ta ışı ve zama ı asıl gidiyor
-----
[28][29] الافة وول ستريت جورنال بمجموعة من القواعد المطبقة على التحرير والنزاعات المتعلقة بمثل هذا المحتوى من بين أسباب هذا الاتجاه [28][29] الافة وول ستريت جورنال بمجموعة من القواعد المطبقة على التحرير والنزاعات المتعلقة بمثل هذا المحتوى من بين أسباب هذا الاتجاه [ ] [ ]

```

**Fig. 5** – Exemples de 10 textes de jeu de données avant et après traitement

## V. Sac de mots (Bag of words)

La variable de sortie du nom de la langue a été convertie en format numérique. Il devrait être de même pour la donnée qui a le format de texte. Pour cela, nous convertissons le texte sous forme numérique en créant un modèle de sac de mots.<sup>2</sup>

### V.1 Exemple illustartif des Sac de mots :

Considérons l'exemple de représentation numérique d'un sac de mots formé de 4 mots qui sont les noms des animaux suivants : chien, chat, poisson et oiseau.

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 texts=["chien chat poisson ", "chien chat chat " , "poisson oiseau oiseau"]
3 cv = CountVectorizer()
4 cv_fit=cv.fit_transform(texts)
5 print(cv.get_feature_names())
6 print(cv_fit.toarray())
```

```
['chat', 'chien', 'oiseau', 'poisson']
[[1 1 0 1]
 [2 1 0 0]
 [0 0 2 1]]
```

Fig. 6 – Exemple illustartif des Sac de mots

**Chaque vecteur est de taille 4. Chaque chiffre est la fréquence d'occurrence du mot de rang considéré dans le texte correspondant.**

---

2. Un modèle de sac de mots, ou BoW, est une façon d'extraire des caractéristiques du texte pour les utiliser dans la modélisation, par exemple avec des algorithmes d'apprentissage automatique. L'approche est très simple et flexible, et peut être utilisée dans une myriade de façons pour extraire des caractéristiques des documents.

Un sac de mots est une représentation du texte qui décrit l'occurrence des mots dans un document. On l'appelle un "sac" de mots, car toute information sur l'ordre ou la structure des mots dans le document est écartée. Le modèle se préoccupe uniquement de la présence de mots connus dans le document, et non de leur emplacement dans le document.<sup>[3]</sup>



## V.2 Application du concept de Sac de mots sur le jeu de données :

On procède le démarche suivant :

```
1 from sklearn.feature_extraction.text import CountVectorizer
2 cv = CountVectorizer()
3 X = cv.fit_transform(data_list).toarray()
4 X.shape
```

De plus, On supprime data\_list pour libérer l'espace mémoire :

```
1 import gc
2 del data_list
3 gc.collect()
```

**Le vecteur du comptage du nombre de mots est de taille 34937.**

**On peut l'augmenter en ajoutant des textes avec de nouveaux mots dans le jeu de données.**

**Il n'est pas possible de visualiser les vecteurs des sacs de mots car leur taille est très élevée. En fait, en utilisant la commande `print(cv.get_feature_names())` le compilateur génère les premiers mots par ordre alphabétique puis le message d'erreur "Output exceeds the size limit" est affiché. Ainsi, la visualisation d'un seul vecteur nécessite son traitement dans un éditeur de texte indépendamment des autres vecteurs!!**

## VI. Entraînement du modèle , prédiction et Evaluation des performances

L'étape suivante consiste à créer l'ensemble d'apprentissage, pour créer le modèle de classification et l'ensemble de test, pour évaluer les performances du classifieur conçu.

A cause de la taille des données relativement importante , on modifie le nombre de bits alloués pour chaque variable, tout en respectant les formats compatibles en Python.

```
1 from sklearn.model_selection import train_test_split
2 import numpy as np
3 y = y.astype(np.int8)
4 X = X.astype(np.int16)
5 x_train, x_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

**Le nombre 0.2 dans la ligne de code précédente correspond au la partition de données de test par rapport au jeu de donnée global, C'est à dire 20% de la data va être conservée pour le test et 80% sera exploitée en training. Ainsi, pour le test on aura une séquence de taille `len(x_test)=2068` et pour l'apprentissage une séquence de taille `len(x_train)=8269`.**

On utilise l'algorithme naive\_bayes avec les données d'apprentissage et on le sauvegarde :

```
1 from sklearn.naive_bayes import MultinomialNB
2 model = MultinomialNB()
3 model.fit(x_train, y_train)
4 import pickle
5 filename = 'language_detection_model.sav'
6 pickle.dump(model, open(filename, 'wb'))
```

L'apprentissage a pris un peu du temps pour être accompli et on peut maintenant prédire (classifier) des données de test et évaluer les performances du modèle à travers la matrice de confusion et l'accuracy :

```
1 y_pred = model.predict(x_test)
2 from sklearn.metrics import accuracy_score, confusion_matrix
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6 ac = 100*accuracy_score(y_test, y_pred)
7 print(ac)
8 cm = confusion_matrix(y_test, y_pred)
9 cm=100*cm/ cm.astype(np.float).sum(axis=1)
10 plt.figure(figsize=(15,10))
11 sns.heatmap(cm, annot = True)
12 plt.show()
```

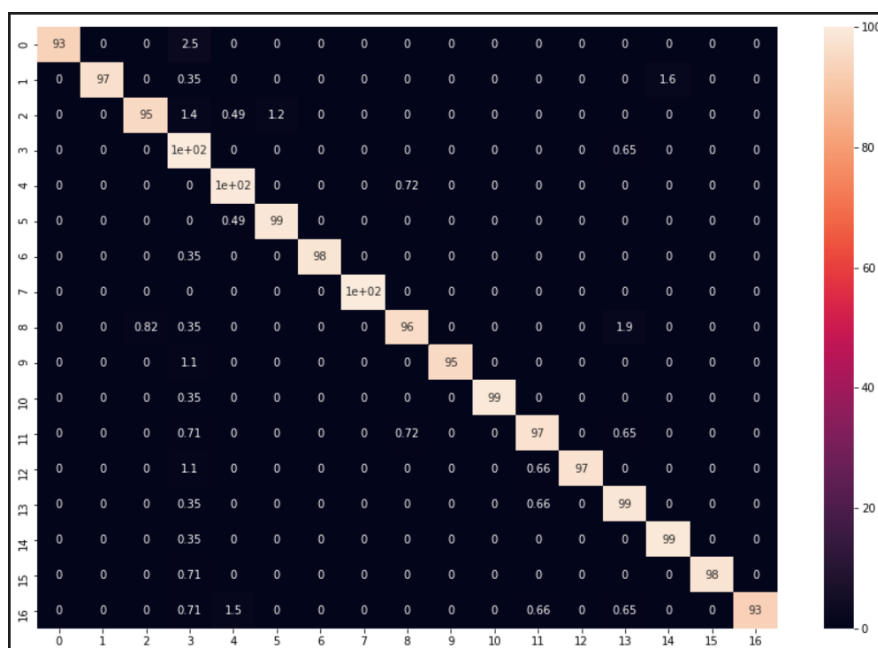


Fig. 7 – Matrice confusion du modèle étudié

**On obtient une précision  $ac=97.87234042553192$  %**

Les performances globales du système de détection des langues semblent être satisfaisantes. On a obtenu une classification parfaite pour les langues {English, French , Hindi } et on remarqué une confusion dans les langues {Arabic , Turkish}

La langue Turkish a été confondue au plus avec la langue française : Ceci est expliquer par la similarité des lettres des 2 langues.

De même la langue Arabe a été confondu au plus avec la langue anglaise.

**Remarque :** Notre modèle peut avoir un problème d'overfitting<sup>3</sup>, on doit vérifier son adaptabilité avec de nouvelles données ce qui l'objectif de la dernière partie.

## VII.Prédiction avec de nouvelles données

Dans cette partie, on teste la prédiction du modèle en utilisant du texte dans différentes langues qui ne faisait pas partie du jeu de données initiales comme expliquer précédemment. Pour cela, on écrit d'abord une fonction de prédiction et on l'applique aux exemples :

```

1 def predict(text):
2     x = cv.transform([text]).toarray()
3     lang = model.predict(x)
4     lang = le.inverse_transform(lang)
5     print("The langauge is in",lang[0])
6     #####

```

```

predict("Est-ce que cet exercice vous a permis d'avoir un aperçu introductif au traitement naturel du langage ?")
predict("Did this exercise give you an introductory overview to natural language processing?")
predict("Bu alıştırma size dođal dil işlemeye giriş niteliğinde bir genel bakış sağladı mı?")
predict("هل أعطاك هذا التمرين نظرة عامة تمهيدية حول معالجة اللغة الطبيعية؟")
predict("¿Este ejercicio le brindó una introducción al procesamiento del lenguaje natural?")
predict("ഇതുകൂറായാമനസംസ്കാരവികാസാഭിവാസാലിക്രിയകൾആമുഖരവലാകുന്നതിൽകിയാ?")
predict("Это упражнение дало вам вводный обзор обработки естественного языка?")

```

✓ 0.5s

```

The langauge is in French
The langauge is in English
The langauge is in Turkish
The langauge is in Arabic
The langauge is in Spanish
The langauge is in Malayalam
The langauge is in Russian

```

**Fig. 8** — Résultat de prédiction des langues des échantillons données

**On obtient une exactitude de prédiction des langues correspondantes aux échantillons des textes.**

3. **Overfitting** : un modèle trop spécialisé sur les données du Training Set et qui se généralisera mal

# Conclusion

J'ai réussi dans ce travail à concevoir un modèle détecteur de langue de rédaction d'un texte en utilisant le concept de traitement naturel du langage (**NLP**) avec l'environnement de travail Python 3.9 en explorant ses différentes bibliothèques qui ont servi à faciliter énormément les tâches à faire pour l'apprentissage machine (**ML**).

Le modèle réalisé a présenté de bonnes performances au niveau de la classification des textes avec un peu de confusion pour les langues similaires. Cette confusion peut être minimisée en entraînant le modèle d'avantage avec d'autres données.

Pour conclure, J'ai pu à travers ce TD machine étudier de différentes étapes de conception d'un modèle de détection de langue de rédaction d'un texte et mettre en oeuvre un modèle qui prédit avec exactitude la langue des différentes données au niveau de la dernière partie du travail.

# Bibliographie

- [1] ODSC Open Data Science. Why you should be using jupyter notebooks. <https://odsc.medium.com/why-you-should-be-using-jupyter-notebooks-ea2e568c59f2>.
- [2] Tout savoir sur la méthode du web scraping. <https://www.laou.fr/conseils-pro/tout-savoir-sur-la-methode-du-web-scraping/#:~:text=L'art%20d'extraire%20des,et%20de%20les%20r%C3%A9utiliser%20ensuite>.
- [3] A gentle introduction to the bag-of-words model. <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>.